第六章练习题

1、选择题

（1）在 C 语言中，引用数组元素时，其数组下标的数据类型允许是（ C ）

A. 整型常量　　B. 整型表达式

C. 整型常量或整型表达式　　D. 任何类型的表达式

（2）若有说明：int a[10];，则对数组元素的正确引用是（D）

A. a[10];　　B. a[3.5]　　C. a(5)　　D. a[10-10]

（3）设有数组定义：char array[]="China";，则数组 array 所占的空间为（ C ）

A. 4 个字节　　B. 5 个字节　　C. 6 个字节　　D. 7 个字节

（4）若二维数组 a 有 m 列，则在 a[i][j]前的元素个数为（ B ）

A. j*m+i　　B. i*m+j

C. i*m+j-1　　D. i*m+j+1

（5）若有说明：int a[][3]={1,2,3,4,5,6,7};，则 a 数组第一维的大小是（ B ）

A. 2　　B. 3　　C. 4　　D. 无法定值

（6）以下不正确的定义语句是（B）

A. double x[5]={2.0, 4.0, 6.0, 8.0, 10.0};

B. int y[5]={0, 1, 3, 5, 7, 9};

C. char c1[]={'1', '2', '3', '4', '5'}

D. char c2[]={'\x10', '\xa', '\x8'}

（7）以下不能对二维数组 a 进行正确初始化的语句是（ C ）

A. int a[2][3]={0};

B. int a[][3]={{1, 2}, {0}};
C. int a[2][3]={{1, 2}, {3, 4}, {5, 6}};
D. int a[][3]={1, 2, 3, 4, 5, 6};

（8）以下能对二维数组 a 进行正确初始化的语句是（ A、B ）
A. int a[2][]={{1, 0, 1}, {5, 2, 3}};
B. int a[][3]={{1, 2, 3}, {4, 5, 6}};
C. int a[2][4]={{1, 2, 3}, {4, 5}, {6}};
D. int a[][3]={{1, 0, 1}, {}, {1, 1, 1, 1}};

（9）以下不能正确进行字符串赋初值的语句是（ A ）
A. char str[5]="good!";
B. char str[]="good!";
C. char str[8]="good!";
D. char str[5]={'g', 'o', 'o', 'd'};

（10）判断字符串 s1 是否大于字符串 s2，应当使用（ D ）
A. if(s1>s2)
B. if(strcmp(s1, s2))
C. if(strcmp(s2, s1)>0)
D. if(strcmp(s1, s2)>0)

（11）给出以下定义，则正确的叙述为（ A ）
char x[ ] = "abcdefg";
char y[ ] = {'a', 'b', 'c', 'd', 'e', 'f', 'g'};
A. 数组 x 和数组 y 等价
B. 数组 x 和数组 y 的长度相同
C. 数组 x 的长度大于数组 y 的长度
D. 数组 x 的长度小于数组 y 的长度

（12）以下程序的输出结果是（ B ）

```
int main()
{
    char st[20] = "hello\0\t\\\";
    printf ("%d %d \n", strlen(st), sizeof(st));
    return 0;
}
```

A. 9 9　　B. 5 20　　C. 13 20　　D. 20 20

（13）定义如下变量和数组：

```
int k;
int a[3][3]={1, 2, 3, 4, 5, 6, 7, 8, 9};
```

则下面语句的输出结果是（A）

```
for (k = 0; k < 3; k++)
        printf ("%d ", a[k][2-k] );
```

A. 3 5 7　　B. 3 6 9　　C. 1 5 9　　D. 1 4 7

（14）当执行下面的程序时，如果输入 ABC，则输出结果是（ A ）

```
#include <stdio.h>
#include <string.h>
int main()
{
```

```
    char ss[10]="1,2,3,4,5";
    gets(ss);
    strcat(ss, "6789");
    printf("%s\n", ss);
    return 0;
  }
```

A. ABC6789　　B. ABC67

C. 12345ABC6　　D. ABC456789

（15）以下程序的输出结果是（　D　）

```
int main()
{
  char w[][10] = {"ABCD", "EFGH", "IJKL", "MNOP"}, k;
  for (k = 1; k < 3; k++)
      printf ("%s\n", w[k]);
  return 0;
 }
```

A. ABCD　　B. ABCD　　C. EFG　　D. EFGH

FGH　　EFG　　JK　　IJKL

KL　　IJ　　O

M

（16）以下程序的输出结果是（　A　）

```
int main()
{
  char arr[2][4];
  strcpy (arr[0], "you");
  strcpy (arr[1], "me");
  arr[0][3] = '&';
  printf ("%s \n", arr);
  return 0;
 }
```

A. you&me　　B. you　　C. me　　D. err

（17）已知：char str1[8], str2[8]={"good"}; 则在程序中不能将字符数组 str2 赋值给 str1 的语句是（ A ）

A. str1=str2　　B. strcpy(str1, str2);

C. strncpy(str1, str2, 6);　　D. memcpy(str1, str2, 5);

（18）下面程序段的运行结果是（C）（□ 代表空格）

```
char c[5] = {'a', 'b', '\0', 'c', '\0'};
printf ("%s", c);
```

A. 'a''b'　　B. ab　　C. abc　　D. ab□

（19）下面程序段的运行结果是（　C　）（代表空格）

```
char a[7] = "abcde";　　char b[4] = "ABC";
```

a[3]='\0'

strcpy (a, b);　　printf (”%c”, a[4]);

A. □　　B. \0　　C. e　　D. f

（20）下面程序段的运行结果是（ C ）

```
int main()
{
  char ch[7] = {”65ab21”};
  int i, s = 0;
  for ( i = 0; ch[i] >= ’0’ && ch[i] <= ’9’; i += 2)
        s = 10 * s + ch[i] – ’0’;
  printf (”%d\n”, s);
  return 0;
 }
```

A. 12ba56　　B. 6521　　C. 6　　D. 62

2、程序填空题

（1）以下是个评分统计程序，共有 8 个评委打分，统计时，去掉一个最高分和一个最低分，其余 6 个分数的平均分即是最后得分，程序最后应显示这个得分，显示精度为 1 位整数，2 位小数，程序如下，请将程序补充完整。

d . f f

```
#include <stdio.h>
int main()
{
  float x[8] = {9.2, 9.5, 9.8, 7.4, 8.5, 9.1, 9.3, 8.8};
  float aver, max, min;
  int i;                    aver = 0
  for ( ____i=0____; i < 8; i++)
        aver += x[i];
  max = ____x[ 0 ]____;
  min = max;
  for ( i = 1; i < 8; i++)
  {
      if (max < x[i])
          max = x[i];
      if (___min>x[ i ]___)
          min = x[i];
  }
  aver = ( aver - min -max ) / 6;
  printf (”Average = %.1f \n”, aver);
  return 0;
}
```

%4.2f

（2）以下程序是实现在 M 行 N 列的二维数组中，找出每一行上的最大值。请将程序补充完整。

```
#define M 3
#define N 4
```

```
int main( )
{
    int x[M][N] = {1, 5, 7, 4, 2, 6, 4, 3, 8, 2, 3, 1};
    int i, j, p;
    for ( i = 0; i < M; i++)
    {
        p = 0;
        for ( j = 1; j < N; j++ )
        {
            if (x[i][p] < x[i][j])
                _____p=j;_____
            printf ("The max value in line %d is %d\n", i, _____p_____);
        }
    }
    return 0;
}
```

x[i][p]

（3）下面程序的功能是在三个字符串中找出最小的。请将程序补充完整。

```
#include <stdio.h>
#include <string.h>
int main( )
{
    int i;
    char s[20], str[3][20];
    for ( i = 0; i < 3; i++)
        gets (str[i]);
    strcpy (s, _____str[0]_____);
    if( strcmp (s, str[2]) > 0)
        strcpy (s, str[2]);
    printf ("The min string is %s\n", _____s_____);
    return 0;
}
```

写成复合语句

strcmp(str[0],str[1])>0?str[1]:str[0]

（4）下面程序的功能是将键盘输入的字符串 str 中的所有'c'字符用'C'替换。请将程序补充完整。

```
#include <stdio.h>
#include <string.h>
int main( )
{
    int i;
    char str[80];
    gets(_____str[ ]_____);
    for ( i = 0; _____i<80_____; i++)
    {
        if( str[i] != 'c' )
            _____;_____
        str[i]='C' ;
    }
```

_str[i]!='\0'

continue ;

```
            else str[i]='C';
        }
        printf ("%s\n", str);
        return 0;
    }
```

3、编程题

1. 用筛选法求 100 之内的素数。
2. 用选择法对 10 个整数排序。
4. 有一个已排好序的数组，要求输入一个数后，按原来排序的规律将它插入数组中。
6. 输出以下的杨辉三角形(要求输出 10 行)。

```
1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
1   5   10  10  5   1
⋮   ⋮   ⋮   ⋮   ⋮   ⋮
```

13. 编一程序，将两个字符串连接起来，不要用 strcat 函数。

15. 编写一个程序，将字符数组 s2 中的全部字符复制到字符数组 s1 中。不用strcpy 函数。复制时，'\0'也要复制过去。'\0'后面的字符不复制。

第七章练习题

1、填空题

函数是一段可重复调用的、功能相对独立完整

（1）在 C 语言程序中，功能模块是由函数来实现的。函数是__主函数__的程序段。

（2）从函数定义的角度看，函数可分为__主函数__和__子函数__两种。标准库函数　自定义函数

（3）对于有返回值的函数来说，通常函数体内包含有__输出__语句，其格式为__return x;__，用于将返回值带给调用函数。
return　　return (表达式)

（4）调用带参数的函数时，实参列表中的实参必须与函数定义时的形参__数目__相等、__类型__相符。

（5）对带有参数的函数进行调用时，参数的传递方式主要有__值传递__和__地址传递__两种方式。

（6）变量的作用域和生存期是从__有效范围__和__存在时间__的角度来体现变量的特性。

（7）变量的存储类型可分为__静态__和__动态__两种类型。C 语言中，对变量的存储类型的说明有以下四种，即__auto__、__static__、__extern__和__register__。

（8）静态局部变量若在定义时未赋初值，则系统自动赋初值__0__。其生存期是__整个源程序__，其作用域是__定义了该变量的源函数__　在定义变量的函数内或复合语句中

（9）C 语言程序中，函数不允许嵌套__自身__，但允许嵌套__另一个函数__。

定义　　调用

2、选择题

（1）以下正确的说法是（　C　）

A. 用户若需要调用标准库函数，调用前必须重新定义

B. 用户可以重新定义标准库函数，若如此，该函数将失去原有含义

C. 系统根本不允许用户重新定义标准库函数

D. 用户若需调用标准库函数，调用前不必使用预编译命令将该函数所在文件包含到用户源文件中，系统自动去调用

（2）以下不正确的说法是（　B　）

A. 实参可以是常量、变量或表达式

B. 形参可以是常量、变量或表达式

C. 实参可以为任何类型

D. 形参应与其对应的实参类型一致

（3）以下正确的函数定义形式是（ A ）

A. double fun(int x, int y)　　B. double fun(int x; int y)

C. double fun(int x, int y);　　D. double fun(int x, y);

（4）以下正确的说法是（ C ）

A. 定义函数时，形参的类型说明可以放在函数体内

B. return 后边的值不能为表达式

C. 如果函数值的类型与返回值类型不一致，以函数值类型为准

D. 如果形参与实参类型不一致，以实参类型为准

以形参为准

（5）在 C 语言中，函数的隐含存储类别是（C）

A. auto　　B. static　　C. extern　　D. 无存储类别

（6）凡是函数中未指定存储类别的局部变量，其隐含的存储类别为（ A ）

A. 自动（auto）　　B. 静态（static）

C. 外部（extern）　　D. 寄存器（register）

（7）若使用一维数组名作函数实参，则以下正确的说法是（ C ）

A. 必须在主调函数中说明此数组的大小

B. 实参数组类型与形参数组类型可以不匹配

C. 在被调用函数中，不需要考虑形参数组的大小

D. 实参数组名与形参数组名必须一致

（8）已有如下数组定义和 f 函数调用语句，则在 f 函数的说明中，对形参数组 array 的正确定义方式为（C）

```
int a[3][4];
f(a);
```

A. f(int array[][6])　　B. f(int array[3][])

C. f(int array[][4])　　D. f(int array[2][5])

（9）若用数组名作为函数的实参，传递给形参的是（ A ）

A. 数组的首地址　　B. 数组第一个元素的值

C. 数组中全部元素的值　　D. 数组元素的个数

（10）函数调用不可以（ C ）

函数返回值在寄存器中，**没有地址**，不能作为形参，但可以作为实参

A. 出现在执行语句中　　B. 出现在一个表达式中

C. 作为一个函数的实参　　D. 作为一个函数的形参

（11）C 语言规定，函数返回值的类型是由（ D ）

A. return 语句中的表达式类型所决定

B. 调用该函数时的主调函数类型所决定

C. 调用该函数时系统临时决定

D. 在定义该函数时所指定的函数类型所决定

（12）C 语言规定：简单变量作为实参时，它和对应形参之间的数据传递方式是（ B ）

A. 地址传递

B. 单向值传递

C. 由实参传给形参，再由形参传回给实参

在C语言中，函数的隐含存储类别是 extern。这意味着函数默认情况下具有外部链接，可以在其他文件中访问。如果没有明确指定存储类型，函数就会被认为是extern类型。

extern 存储类别

当函数被定义为extern时，**它可以在其他文件中被调用和访问。**

例如，以下两种定义是等价的：

```
void func();
extern void func();
```

这表示函数func可以在定义它的文件之外的其他文**件中使用。**

static 存储类别

如果函数被定义为 static，则其存储类别为静态。这意味着该函数只能在定义它的文件中访问，**不能在其他文件中使用。**

例如：

```
static void func();
```

这种定义方式限制了函数的作用范围，**使其只能在当前文件中使用**

D. 由用户指定的传递方式

（13）以下只有在使用时才为该类型变量分配内存的存储类型说明是（ B ）

A. auto 和 static　　B. auto 和 register

C. register 和 static　　D. extern 和 register

（14）以下叙述中不正确的是（ C ）

A. 在不同的函数中可以使用相同名字的变量

B. 函数中的形式参数是局部变量

C. 在一个函数内定义的变量只在本函数范围内有效

D. 在一个函数内的复合语句中定义的变量在本函数范围内有效

（15）有以下程序，程序运行后的输出结果是（ D ）

```
float fun (int x, int y)
{ return (x + y); }
int main()
{
   int a = 2, b = 5, c = 8;
   printf ("%3.0f\n", fun ((int) fun (a+c, b), a － c));
   return 0;
 }
```

A. 编译出错　　B. 9　　C. 21　　D. 9.0

（16）下列程序执行后的输出结果是（ C ）

```
char st[ ] = "hello, friend!";
void func1 (int i)
{
   printf ("%c", st[i]);
   if (i < 3) { i += 2; func2 (i); }
 }
void func2 (int i)
{
   printf ("%c", st[i]);
   if (i < 3) { i += 2; func1 (i); }
 }
int main ()
{
   int i = 0; fun1 (i); printf ("\n");
   return 0;
 }
```

h l o
i 0 2 4

A. hello　　B. hel　　C. hlo　　D. hlm

（17）有以下程序，程序运行后的输出结果是（ B ）

```
int f (int n)
{
   if (n == 1)     return 1;
   else     return f (n -1) + 1;
```

```
}
int main( )
{
  int i, j = 0;
  for (i = 1; i < 3; i++)        j += f (i);
  printf (”%d\n”, j);
  return 0;
}
```

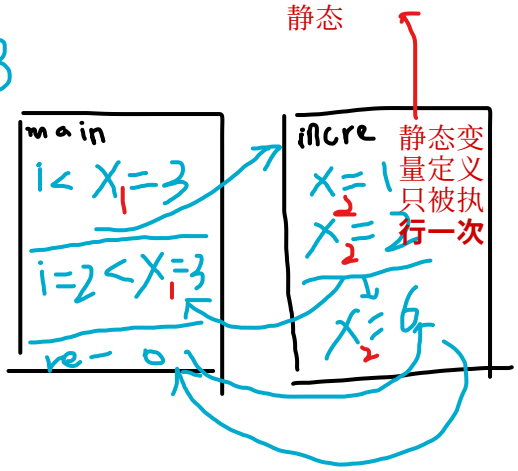
A. 4　　B. 3　　C. 2　　D. 1

该子函数声明的x是几仅作用于自己的静态

（18）以下程序的输出的结果是（ C ）

```
void incre ();
int x = 3;
int main ()
{
  int i;
  for (i = 1; i < x; i++)        incre ();
  return 0;
}
void incre ()
{
  static int x = 1;
  x *= x + 1;
  printf (” %d ”, x);
}
```

main的X是全局变量



A. 3 3　　B. 2 2　　C. 2 6　　D. 2 5

（19）以下程序的输出的结果是（ C ）

```
int a = 3;
int main ()
{
  int s = 0;
  { int a = 5; s += a++; }
  s += a++; printf (”%d\n”, s);
  return 0;
}
```

A. 8　　B. 10　　C. 7　　D. 11

（20）以下程序的输出的结果是（ C ）

```
void f (int a[ ], int i , int j)
{
  int t;
  if (i < j) { t = a[i]; a[i] = a[j];        a[j] = t;     f (a, i+1, j-1);
}
int main()
{
```

```
    int i , a[5] = {1, 2, 3, 4, 5};
    f (a, 0, 4);
    for ( i = 0; i < 5; i ++) printf (”%d, ”, a[i]);
  }
  }
```

A. 5, 4, 3, 2, 1

B. 5, 2, 3, 4, 1

C. 1, 2, 3, 4, 5

D. 1, 2, 3, 4, 5